UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/715,782	11/18/2003	Dwayne Need	MFCP.110238	4543

45809 7590 07/25/2006

SHOOK, HARDY & BACON L.L.P.
(c/o MICROSOFT CORPORATION)
INTELLECTUAL PROPERTY DEPARTMENT
2555 GRAND BOULEVARD
KANSAS CITY, MO 64108-2613

EXAMINER	
PATEL, MANGLESH M	

ART UNIT	PAPER NUMBER
2178	

DATE MAILED: 07/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/715,782	NEED ET AL.	
	Examiner	Art Unit	
	Manglesh M. Patel	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This **FINAL** action is responsive to the amendment filed on 05/09/06.

2. Claims 1-32 are pending. Claims 1, 13, 25 and 30 are independent claims.

Withdrawn Rejections

3. The 35 U.S.C. 103(a) rejection of claim 1-32 with cited reference of Matthews U.S. 6,259,435 in view of Adler U.S. 5,812,818 further in view Knauft U.S. 6,981,217 have been withdrawn in light of the amendment.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Atkins (U.S. Pub 2004/0181776, filed Mar 13, 2003).

Regarding Independent claims 1 and 25, Atkins discloses a computerized method for processing a user input event having code associated therewith, said method comprising: Receiving notification of said input event, said notification including the associated code (abstract, fig 4 & paragraphs 8-11, wherein the keyboard hook module sends the keystroke to the keystroke conversion module, therefore input events with associated codes are received); Determining whether a text converting component is interested in performing a conversion action with respect to said input event (paragraph 37, wherein the keyboard hook module receives keyboard events from an input interface of the IME. The text converting component must be active for the keystroke conversion module to operate or perform the correct conversion, in this case being Unicode); Notifying an application of said input event by providing said application an obfuscation of said code when the text converting component is interested in performing said conversion action with respect to said input event (abstract, fig 4, paragraphs 8-11 & paragraphs 33-37, wherein the active application is notified by setting flags, in the case where the application does not support Unicode then a obfuscated version of the input is received by the application. This obfuscated version is simply the non-unicode input. The text-converting component is interested in performing the conversion because the IME includes the

keyboard conversion module with a focus hook module. The focus hook module determines the currently active application and weather or not it supports unicode. The text converting component is always active because it is always sending the keyboard input to the conversion module, however if the active application cannot support the input then a obfuscated or non-unicode say ASCII value is sent to the active application instead). Atkins determines the active application and its support for unicode, if it does not support unicode then an obfuscated version is sent, meaning not the actual unicode value. However if it does support unicode then the Unicode value is stored in temporary storage and then sent to the active unicode supported application. Atkins does not explicitly teach obfuscation of code at all times to an application when the IME is active. However it would have been obvious to one of ordinary skill in the art to prevent applications with no unicode support from receiving the converted input from the IME. The motivation for doing so would have been to prevent the application from receiving converted input it cannot support. The text converting component must be active for the conversion to take place, and the application having no support for that conversion receives the normal keyboard input whereas the converted input is obfuscated from the active application has determined by the focus hook module.

Regarding Dependent claim 2, which depends on claim 1, Atkins discloses wherein said user input event is communicated via a keyboard, a screen with user input capability, a mouse, and/or a device with voice input capacity (fig 1-2 & 4, wherein the input event includes a keyboard 106, screen 104 and a mouse 110).

Regarding Dependent claims 3, 15 and 26, Atkins discloses wherein said code identifies at least a portion of a letter, a character, an ideograph or a symbol associated with said user input event (paragraphs 8-11, wherein the keyboard conversion module includes identifying a code associated with the input event. The code includes a character, letter and symbols based on the character set selected by the user described in paragraph 31).

Regarding Dependent claim 4, which depends on claim 1, Atkins discloses communicating said code to the application when the text converting component is not interested in processing said user input event (paragraphs 8-11, wherein when the keyboard, focus and conversion modules are not operating on the machine then any input event and associated code is directly sent to the active application, such as an ASCII value of the keyboard input).

Regarding Dependent claims 5 and 18, Atkins discloses wherein the text converting component is configured to convert said code to a standard for coding text (paragraphs 8-11, wherein the unicode conversion is a standard for coding text).

Regarding Dependent claims 6 and 19, Atkins discloses wherein said standard is Unicode (paragraphs 8-11).

Regarding Dependent claim 7, which depends from claim 1, Atkins discloses revealing said code to the application in response to a request to disclose said code (paragraphs 8-11, wherein when Unicode is supported then the code is revealed to the active application).

Regarding Dependent claim 8, which depends on claim 1, Atkins discloses determining whether a computer component is interested in processing said input event (paragraphs 8-11, wherein the focus hook module is responsible for determining if the active application is interested in processing input event).

Regarding Dependent claims 9, 24, 28 & 32, which depends on claim 8, Atkins discloses obfuscating said code from an application when the computer component is interested in processing input event (abstract, fig 4, paragraphs 8-11 & paragraphs 33-37, wherein the active application is notified by setting flags, in the case where the application does not support Unicode then a obfuscated version of the input is received by the application. This obfuscated version is simply the non-unicode input. The text-converting component is interested in performing the conversion because the IME includes the keyboard conversion module with a focus hook module. The focus hook module determines the currently active application and weather or not it supports unicode. The text converting component is always active because it is always sending the keyboard input to the conversion module, however if the active application cannot support the input then a obfuscated or non-unicode say ASCII value is sent to the active application instead). Atkins determines the active application and its support for unicode, if it does not support unicode then an obfuscated version is sent, meaning not the actual unicode value. However if it does support unicode then the Unicode value is stored in temporary storage and then sent to the active unicode supported application. Atkins does not explicitly teach obfuscation of code at all times to an application when the IME is active. However it would have been obvious to one of ordinary skill in the art to prevent applications with no unicode support from receiving the converted input from the IME. The motivation for doing so would have been to prevent the application from receiving converted input it cannot support. The text converting component must be active for the conversion to

take place, and the application having no support for that conversion receives the normal keyboard input whereas the converted input is obfuscated from the active application has determined by the focus hook module.

Regarding Dependent claim 10, which depends on claim 9, Atkins discloses notifying the application that the computer component is interested in processing said input event (paragraphs 8-11).

Regarding Dependent claims 11, 23 and 29, Atkins discloses wherein said computer component is an input method editor (paragraphs 8-11).

Regarding Dependent claim 12, which depends on claim 8, Atkins discloses
wherein said computer component is configured to allow a user to enter at least a portion of a letter, a character, an ideograph or a symbol associated with a desired language (paragraphs 8-11).

Regarding Independent claim 13, Atkins discloses a computer system for processing a user input event having code associated therewith, the system comprising: One or more text converting components (paragraphs 8-11, wherein the keyboard conversion module represents a converting component); One or more applications (paragraphs 8-11, wherein the focus hook module determines an active application, thereby including one or more applications that may or may not support unicode); An input manager configured to interact with said one or more text converting components and said one or more applications, wherein said input manager is configured to receive notification of an input event, said notification including the associated code, and wherein said input manager is further configured to prevent said one or more applications from handling said input event by obfuscating said code from the one or more applications when said one or more text converting components are interested in performing a conversion with respect to said input event (paragraphs 8-11 & paragraph 31, wherein the configuration interface includes a keyboard hook module thereby including an input manager that interacts with the conversion module which represents a text converting component. The notifications are received by input manager that include the code from the conversion module and the focus hook module which prevents one or more active applications from receiving the input event by obfuscating the input to prevent sending an input event which the application cannot support, the obfuscated version being the regular keystroke ASCII value instead of the UNICODE value). Atkins determines the active application and its support for unicode, if it does not support unicode then an obfuscated version is sent, meaning not the actual unicode value. However if it does support unicode then the Unicode value is stored in temporary storage and then

sent to the active unicode supported application. Atkins does not explicitly teach obfuscation of code at all times to an application when the IME is active. However it would have been obvious to one of ordinary skill in the art to prevent applications with no unicode support from receiving the converted input from the IME. The motivation for doing so would have been to prevent the application from receiving converted input it cannot support. The text converting component must be active for the conversion to take place, and the application having no support for that conversion receives the normal keyboard input whereas the converted input is obfuscated from the active application has determined by the focus hook module.

Regarding Dependent claim 14, which depends on claim 13, Atkins discloses wherein said code is generated by a driver associated with an input device (paragraphs 8-11, wherein the keyboard includes a driver, the driver includes the keyboard hook and conversion modules).

Regarding Dependent claim 16, which depends on claim 13, Atkins discloses wherein said user input event is communicated via an input device that is not configured according to a desired language (paragraphs 8-11, wherein the keyboard is the input device. The device itself is not configured for a desired language. The input manager includes the character set as part of the configuration interface for selecting a desired language).

Regarding Dependent claim 17, which depends on claim 13, Atkins discloses wherein said input manager is further configured to communicate said code to one or more applications when none of the text converters are interested in processing said user input event (paragraphs 8-11, when the modules are not active then the regular keystroke input and its ASCII value are sent to the active application).

Regarding Dependent claim 20, which depends on claim 13, Atkins discloses wherein said input manager is further configured to notify the one or more applications that at least one of said text converting components is interested in performing a conversion action with respect to said input event (paragraphs 8-11, wherein the focus hook module notifies the application sending a flag to it when the text converting component performs a conversion. The flag indicating weather the application supports that type of conversion).

Regarding Dependent claim 21, which depends on claim 13, Atkins discloses wherein said input manager is further configured to reveal said code to one or more of said applications in response to a request to disclose said code (paragraphs 8-11, wherein when Unicode is supported then the code is revealed to the active application).

Regarding Dependent claim 22, which depends on claim 13, Atkins discloses one or more computer components (fig 1-4).

Regarding Dependent claim 27, which depends on claim 25, Atkins discloses a computer component interface component for determining whether one or more computer components are interested in handling said user input event (paragraphs 8-11, wherein the focus hook module determines if the components are interested in handling the unicode input).

Regarding Independent claim 30, Atkins discloses a computer system for processing a user input event having code associated therewith, the system comprising: Means for receiving notification of a user input event having code associated therewith, said notification including the associated code (paragraphs 8-11, wherein the keyboard hook and conversion modules include receiving notification of a user input event with code); Means for converting said code to a value indicating a character or a symbol (paragraphs 8-11, wherein code conversion to a value defined in a character set that includes symbol, characters etc. is performed by the keyboard conversion module); One or more applications (paragraphs 8-11, wherein the focus hook module determines an active application, thereby including one or more applications that may or may not support unicode); Means for interacting with said one or more applications and said converting means in response to notification of said user input event, wherein said means for interacting are configured to prevent one or more applications from handling said user input event by obfuscating said code from the one or more applications when said converting means are interested in performing a conversion action with respect to said input event (paragraphs 8-11 & paragraph 31, wherein the interacting means is performed by the input devices mouse/keyboard that include converting means based on the selection from the configuration interface that is the input manager and consists of the different modules. Wherein the focus hook module is configured to prevent an application from receiving the input event with code when it is determined to not have unicode support). Atkins determines the active application and its support for unicode, if it does not support unicode then an obfuscated version is sent, meaning not the actual unicode value. However if it does support unicode then the Unicode value is stored in temporary storage and then sent to the active unicode supported application. Atkins does not explicitly teach

obfuscation of code at all times to an application when the IME is active. However it would have been obvious to one of ordinary skill in the art to prevent applications with no unicode support from receiving the converted input from the IME. The motivation for doing so would have been to prevent the application from receiving converted input it cannot support. The text converting component must be active for the conversion to take place, and the application having no support for that conversion receives the normal keyboard input whereas the converted input is obfuscated from the active application has determined by the focus hook module.

Regarding Dependent claim 31, which depends on claim 30, Atkins discloses further comprising means for editing an input method (paragraph 31, wherein the IME represents an editing tool).

It is noted that any citation [[s]] to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. [[See, MPEP 2123]]

Response to Arguments

6. Applicant's arguments filed 05/09/06 have been fully considered but are moot in view of the new ground of rejections.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Manglesh M. Patel whose telephone number is (571) 272-5937. The examiner can normally be reached on M, W 6 am-3 pm T, TH 6 am-2pm, Fr 9am-6pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen S. Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Manglesh M. Patel
Patent Examiner
July 14, 2006

CESAR PAULA
PRIMARY EXAMINER